THE MONTE CARLO METHOD: AN ALTERNATIVE METHOD OF NUMERICAL INTEGRATION

Karen A. Haas
Capstone Project
Spring 1985
Advisor: Dr. L. Sons

In the world of mathematics, one of the more fundamental ideas that exists is the idea of a function, mathematically described as a correspondence that associates with each element x, of a set X, a unique element y, of a set Y. This concept is basic to many of the applications of higher mathematics, as are the concepts of integrating a function and differentiating a function. It is the former concept which shall be dealt with here.

Many different, widely used techniques of integration can be found from a variety of sources. Aside from the traditional method of integration learned in any Calculus class, there exist numerical methods of integration which approximate the value of the integral. One simple example, known as the Trapezoidal Rule, evaluates a function on an interval (a,b) at each of the endpoints of the interval, adds these values together, and multiplies this sum by the width of the interval divided by two (that is, the approximation is (f(a)+f(b))*((b-a)/2)). Other such numerical integration techniques exist also, but most of them are variations of the Trapezoidal Rule, designed for more accuracy. However, one other method exists which is not a variation on this method. This alternative form of integration is called the Monte Carlo method, and it shall now be discussed in some detail.

The Monte Carlo method of numerical integration relies heavily on random numbers and probability and at first may

seem an unlikely procedure for integration. However, under the right conditions, especially large enough sample sizes, this method can be competitive and even superior to the traditional numerical techniques already mentioned, especially because it does not involve many difficult computations.

Monte Carlo methods basically developed out of a need to simulate a real life situation. During World War II, physicists needed to analyze and describe the behavior of neutrons in various materials, to get an idea of how to construct shielding devices for nuclear bombs and reactors. To do actual experiments using nuclear bombs would not only be extremely costly, but also extremely dangerous. So the scientists developed a method, using some information that they already possessed about neutrons, to simulate separate events (that is, how certain neutrons would behave) and then to combine this information to figure out how the whole system would behave. The method with which they computed how the whole system would behave relied heavily on random numbers and probability, and hence the Monte Carlo method, named after the European gambling center because of the randomness of the method, was born.

One simple example of the Monte Carlo method, to better understand how the random numbers come into play, is the lottery example. In this example each lottery ticket contains one letter: W, I, or N. Among all of the tickets, 50% of them contain a W, 40% contain an I, and 10% contain an N.

The problem is, how many tickets are needed to spell "WIN"? The first step in the solution to solving the problem is to assign random digits, 0-9, to each of the probabilities of the letter occurring. Since 50% of the tickets have a W, assign 0-4 to the letter W. Then assign 5-8 to the letter I, and finally assign the digit 9 to the letter N. Now, using a random digit table, a correspondence can be made with the numbers and letters as follows. (Random digits taken from the table in the UMAP Module.)

78895	96529	26425	94164	79378	85802	35855	47916	07173	33690
IIINI	NIIWN	WIWWI	NWWIW	INWII	IIIWW	WIIII	WINWI	WIWIW	WWINW
61302	09781	24426	50261	49587	09675	11506	48489	86292	32713
IWWWW	WNIIW	WWWWI	IWWIW	WNIII	WNIII	WWIWI	WIWIN	IIWNW	WWIWW

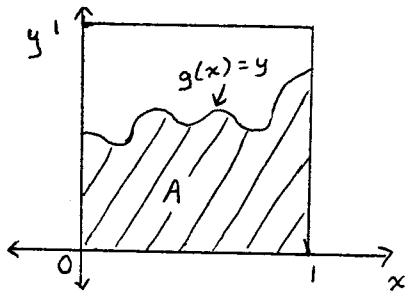
For this very limited example, it takes an average of 8.5 tickets to get the winning combination "WIN". For this example, the random factor is what makes the method work and distinguishes it from other methods. Though the idea may not seem logical at first, it is exactly this randomness that allows the Monte Carlo method to be used in the simulation of numerical integration.

At first glance, integral Calculus and probability and random digits do not seem to have much in common. However, as shall be shown, a Monte Carlo method can not only be useful in evaluating an integral--it can even be competitive or superior to the traditional method of numerical integration, such as the aforementioned Trapezoidal Rule and its relatives, with sufficiently large sample sizes.

Two of the simpler and therefore more common methods

of Monte Carlo integration are the "Hit or Miss" method and the "Sample Mean" method, both of which shall be explained in some detail.

The "Hit or Miss" method of integration estimates the integral $\int_0^1 g(x)dx$ where g is a piecewise continuous function (that is, the interval (0,1) can be partitioned into subintervals such that g is continuous on each of the subintervals). For example, take a function g as in the picture.



Now, letting $A=\{(x,y): y<g(x)\}$, that is A is the area under the curve y=g(x) for x in (0,1), we can see that $\int_0^1 g(x)\,dx=$ {the area under y=g(x) for x in (0,1)} = the area of A. Now, letting k and h be any random numbers on the interval (0,1), then for any different A, which will depend on the g, the probability that the random point (k,h) will lie in the region A is equal to the area of A. In other words, if the area of A comprises 90% of the unit square, then the probability that a random point (k,h) will lie in the area of A is .9, or

$$P_{k,h}(A) = \text{area } (A)$$

where P is the probability of the event. Then, for n pairs of random numbers, (k,h), let B denote the number of these pairs that are members of the area A. Then the "hit or miss"

estimator of the integral is B/n, or in other words, B/n is an estimate for the area of $A=\int_0^1 g(x)dx$. Since for large sets of numbers, the relative frequency of an event converges to the probability of the event, this method is a way to estimate the integral using probability. So, for large n, B/n converges to the area of A. The algorithm used in the computer program is as follows.

Input--n pairs of random numbers (k_i,h_i), i=1,n
(generated through a pseudo-random number generator function)

Say the pairs of numbers (k_i,h_i) is in A if $h_i \leq g(k_i)$, letting B denote how many pairs of (k_i,h_i) are in A.

Output--B/n as the estimate for $\int_0^1 g(x)dx$.

The error term of the "hit or miss" method, derived through probability methods, turns out to be

$$E(\text{ error}^2\ ) \leq 1/4n$$

that is, the expectation of the squared error is less than or equal to 1/4 of the inverse of the sample size. (see Yakowitz pp. 194-195 for computation of the error term) While this may not be very accurate at all for small sample sizes, for a sample of say 100, the "hit or miss" method should be accurate up to one decimal place (also assuming the numbers are truly random). The accuracy can then obviously be increased by increasing the sample size, until the desired accuracy is achieved. Increasing the accuracy of this method is also far easier than increasing the accuracy

of a method such as the Trapezoidal Rule, whose accuracy is increased by creating more complicated functions rather than just a larger sampling. So in this respect, although the Trapezoidal Rule may initially have a better accuracy, it is not difficult to increase the accuracy of the "hit or miss" method, a simpler technique.

Another method of Monte Carlo integration is called the "Sample Mean" method, and it uses a slightly different approach than the "hit or miss" method. Again, the integral to be evaluated is $\int_0^1 g(x)dx$. This method deals more directly with probability and expectation; the method basically says that if k is a random variable having a finite mean, E(k), and R is computed using independent random observations of k by adding the observations together and dividing by n, the number of observations, then R is called the "sample mean" estimator of $\int_0^1 g(x)dx$. In other words, with n random numbers, (k_i), i=1,n, $\sum_{i=1}^{n} g(k_i)/n$ =R= the estimate of $\int_0^1 g(x)\,dx$.
The algorithm used in the program is as follows.

Input--n random numbers (k_i), i=1,n (generated through a pseudo-random number generator function)

Compute $R=(1/n)(\sum_{i=1}^{n} g(k_i))$

Output--R as the estimate for $\int_0^1 g(x)dx$

The error term for the "sample mean" method, again derived through probability methods, is

$$E(\text{ error}^2) = (\int_0^1 (g(x))^2 dx - I^2)(1/n)$$

(see Yakowitz pp.195-196 for computation of the error term).

This method's error term is always less than or equal to the error of the "hit or miss" method, so it is expected that the "sample mean" method will usually be more accurate than the "hit or miss" method, and also that the "sample mean" method is even more competitive with more traditional types of numerical integration.

In the computer program, both of these methods were used on the functions:

function 1-- $4*\sqrt{1-x^2}$

function 2-- sin x

function 3-- x^3

function 4-- e^x

In addition to these single variable functions, two variations of the first function were integrated, namely

function 5-- $4*\sqrt{1-x^2-y^2}$

function 6-- $4*\sqrt{1-x^2-y^2-z^2}$

where multivariate Monte Carlo methods could be tested.

The actual results of the four integrated functions are:
(to 5 decimal places of accuracy)

function 1= 3.14159

function 2= .45970

function 3= .25000

function 4= 1.71828

The results of the computer program are:
(to 5 decimal places of accuracy)

Sample size	function 1	function 2	function 3	function 4
50	3.0800	.53000	.21000	1.8756
	3.1717	.43792	.24378	1.6878
100	3.1200	.50000	.22500	1.7261
	3.0584	.47658	.27900	1.7602
250	3.1440	.46200	.23400	1.6418
	3.0880	.47094	.26871	1.7458
500	3.1520	.45200	.22900	1.6500
	3.1061	.46565	.26282	1.7351

(where the top number represents the "hit or miss" estimator and the bottom number represents the "sample mean" estimator) These values are actually an average of two separate evaluations, in order to try to obtain more accuracy. For the most part, when the sample size is sufficiently large, the values are getting closer to what the actual value of the integral is, although for the "hit or miss" method in function 3, the accuracy needs to be improved a bit. However, the values are relatively close to the actual values, and as is the case of the "hit or miss" method of function 2, the values are obviously converging to the correct answer, which demonstrates how much sample size effects the results. Another effect on the accuracy of the results is the randomness of the numbers. Unfortunately, the supposedly random number generating function is not so random, because it causes the same values to be generated every time the function is envoked. Therefore, the same numbers were used for every computation, except for the last numbers which were only used for the larger sample sizes. Therefore, this computer program, while it

does give an honest representation of how the method works, is not truly representative of the actual results which would be achieved using truly random numbers. However, the idea is there, and it is only a matter of generating random numbers to get results of the correct accuracy.

The evaluations of the multivariate functions, unfortunately did not turn out the same accuracy as did the single variable functions, but this seems largely to be a result of the numbers not being truly random and also not being able to take quite as large of a sample size. A restriction in the program exists where only 2000 separate numbers may be generated, and these numbers are the same for every invocation of the random number generating function. Therefore, these numbers are not accurate at all, although they all do tend to converge, which is an indication that at least the method is working accurately--it is just the randomness that seems to be throwing off the computations.

It should be noted that the "hit or miss" method may be used for functions whose maximum value on (0,1) exceeds 1, however the function must be modified. If the maximum value of the function is M, then apply the "hit or miss" method to the function h defined by $h(x)=(1/M)g(x)$, and then in the final step, multiply the answer obtained by M. (This was done for both function number 1 and function number 4).

In using Monte Carlo techniques, there is a very useful method for evaluating difficult integrals (such as function 1 which requires the special form of a^2-x^2). Now, taking the idea one step further, it can be applied to a probabilistic

model where the method not only uses probability--it also simulates the probability model. The Monte Carlo methods shall then be applied to a probability distribution called the Weibull distribution.

The Weibull model is fairly recent, having been developed and researched only in the past 35 years. It is a probability density function which has been used almost exclusively as a general failure model, and it is given by the formula

$$g(t;\lambda,\gamma)= \begin{cases} \lambda\gamma t^{\gamma-1} e^{-\lambda t^{\gamma}} & \lambda>0,\ \gamma>0,\ t\geq 0 \\ 0 & \text{elsewhere} \end{cases}$$

where and are constants that are dictated by the model, and t denotes time.

One example of the use of a Weibull distribution as a general failure model is the modeling of the failure of a certain electronic device, based on past experience. The Weibull function can give a good approximation to the average failure time of the device, or how long the device will work. This type of information could be quite useful in the development of new devices and the improvement of old devices.

Another example where the Weibull distribution could prove useful is in dispensing medication, such as aspirin. The time it takes for relief from, say, a headache to occur would be quite useful to aspirin manufacturers who need safe yet effective products. A Weibull model would just naturally fit into this scheme, since it is so accurate at measuring a failure rate.

An example of a worked Weibull distribution shall now be presented, as well as the results of Monte Carlo

simulation on the model. The example is as follows (from Shapiro and Gross, p. 294)

A two parameter Weibull distribution will be used in a simulation study to model the time-to-death of a given insect in the study of an ecosystem involving the wildlife in a given area. Scientists believe that 90% of the insects live at least 15 days and that only 10% of them live past 60 days. From this information, determine the average lifetime of the insect.

In order to solve this equation in the most efficient manner, an equation called the median failure time for the Weibull distribution will be solved, rather than the previously given formula. (The median failure time equation is a variation of the other model which adapts quite nicely to computer programs.) The formula is given by

$$t_p=(\lambda^{-1}\ln(p^{-1}))^{\frac{1}{\gamma}}$$

where t_p is the time for which the probability is p that the component has not failed (or in this case that the insect has not died). First the equation must be solved to obtain the values of the parameters λ and γ. Setting $p^{-1}=.9$ and .1 and using $t_{.1}=15$ and $t_{.9}=60$ (where $t_{.9}$ corresponds to $p^{-1}=.1$ and $t_{.1}$ corresponds to $p^{-1}=.9$) gives the system of equations

$$15=((-1/\lambda)\ln(.9))^{\frac{1}{\gamma}} \quad \text{and}$$

$$60=((-1/\lambda)\ln(.1))^{\frac{1}{\gamma}}$$

Solving for λ and γ, gives

$$\frac{60=(-\frac{1}{\lambda}\ln(.1))^{\frac{1}{\gamma}}}{15=(-\frac{1}{\lambda}\ln(.9))^{\frac{1}{\gamma}}} \Rightarrow 4=\left(\frac{2.30258051}{.10536052}\right)^{\frac{1}{\gamma}}$$

$$\Rightarrow \ln(4)=\frac{1}{\gamma}\left(\ln\left(\frac{2.30258051}{.10536052}\right)\right) \Rightarrow \gamma=2.22499241 \approx 2.225$$

Then, plugging in the λ value gives

$$\gamma = .00025461$$

Now the equation for the mean failure time is

$$t_p=((-1/.00025461)\ \ln(p^{-1}))**(1/2.225)$$

which, when integrated, should yield the average life of the insect in the ecosystem.

Using the "sample mean" method of Monte Carlo integration gives the average life of the insect to be approximately 54 days, which is perfectly reasonable as it is within the bounds of the usual lifespan set by the scientists. This shows that Monte Carlo methods may be used for many different applications and that it has many different uses.

As has been shown, the Monte Carlo methods can prove to be useful tools in approximating integrals. While the sample size must be very large in order to get an accurate (four or more decimal places) approximation, samples of smaller size may be used just to get a feel for the value of an integral. For small sample sizes, the Monte Carlo methods provide a ball park figure for the value, which is not as accurate as would be desired, but nevertheless can be used as some sort of approximation.

It is unfortunate that the supposedly random number generating function turned out to not really be random, because

that destroyed the accuracy of the outcomes quite a bit. However, even with the numbers being only pseudo-random, the approximations were rather good for such small sample sizes. Apparently, then, the Monte Carlo methods of integration can offer a viable alternative to traditional and/or often complicated methods of integration. For example, for function 1, the approximation using the Trapezoidal Rule yields the value 2, whereas the Monte Carlo methods are both accurate to the first digit even with the smallest sample size. Though Monte Carlo methods will never replace other methods of numerical integration, maybe someday they will be widely accepted and placed together with the other methods as a viable alternative.

REFERENCES

Shapiro, Samuel S., and Gross, Alan J. Statistical Modeling Techniques. New York: Marcel Dekker, INC., 1981.

Yakowitz, Sidney J. Computational Probability and Simulation. Reading,Massachusetts:Addison-Wesley Publishing Company. 1977

UMAP Module . Unit 269. Monte Carlo: The Use of Random Digits to Simulate Experiments. Dale T. Hoffman. Lexington, Massachusetts. 1983.